

SQL (Basic)

Vanessa Casillas

Graduate Fellow

Before we start

- Download
 - DbeaverSqliteGUI environment
 - <https://dbeaver.io/>
 - Download and install Community version on local machine (installer available for Windows, Mac and Linux)
- Survey
- <https://github.com/CMC-QCL/SQL>

Agenda

SQL Overview

Relational Databases

Databases

Basic SQL Statements

SQL Database Query Difference

Basics SQL Commands Hands-on

Operation Order

Resources

Contact Info

If you want
to...

See data

Relationship between data

Find any cell

Add data to cells

To add order to information

SQL Overview

Structured Query
Language

Crud: create, read,
update, delete

- Create databases and tables
- Look at specific data
- Make changes to data and remove data
- And more (level 2)

Relational Databases

A collection of tables

- Set of columns in each table and rows with data
- Each row is called a **record**
- Data between tables can be related between each other

Collection of tables is
called **schema**

Relational DBMS

Courses_Students

ID	ClassID	Semester
71225	1005	Fall21
86634	1006	Spr22
32238	1009	Spr22

School_Courses

ClassID	Title	ClassNum
1005	Intro to Art History	500
1006	Intro to SQL	501
1009	Intro to Datebases	300

Students_Attendees

ID	Name	Grade	DOB
71225	Lili	Freshman	03/12/1995
32238	Brenda	Senior	05/28/1989
86634	James	Freshman	09/20/1999

The Relational Model

Primary Key

↓

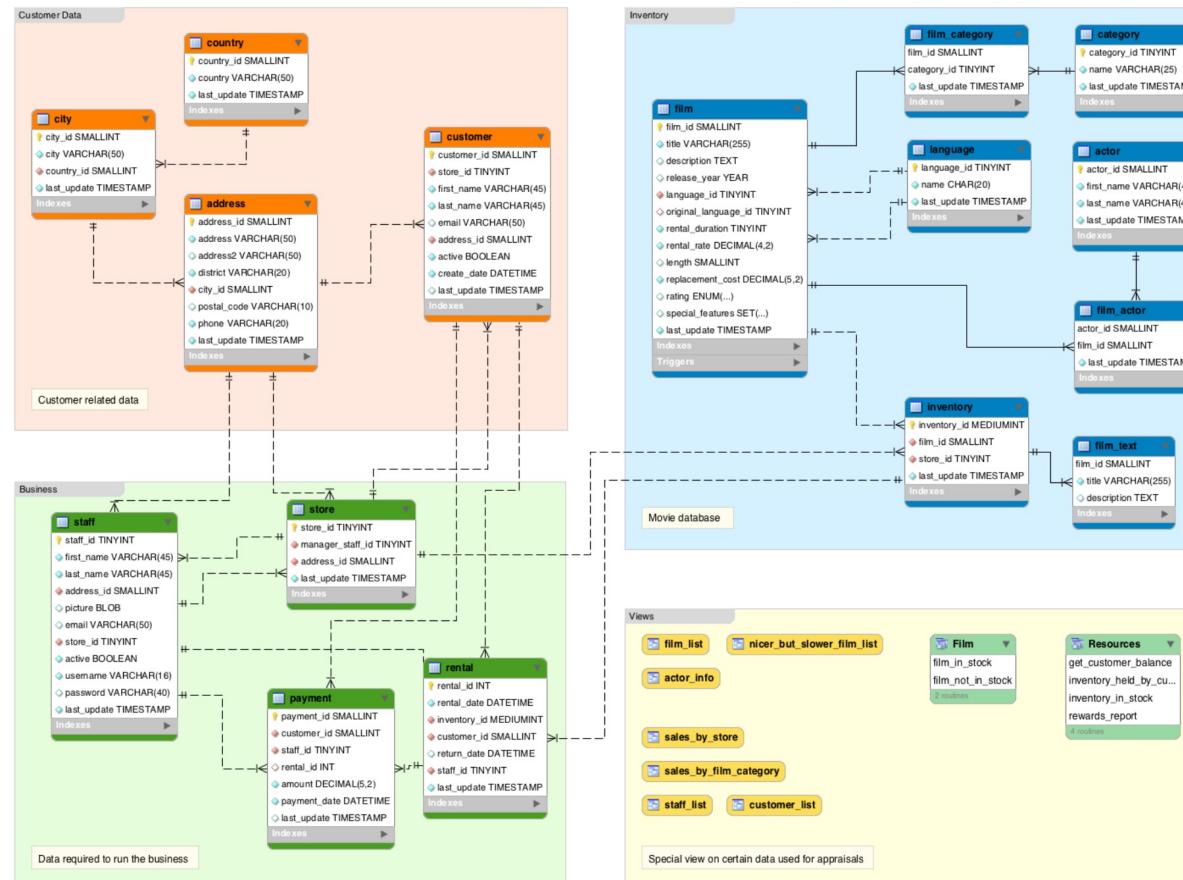
Row or Tuple →

CustomerID	CustomerName	Status
1	Instagram	Active
2	Facebook	Active
3	Myspace	Inactive

Rows are cardinality
Columns are degree
Table called a Relation

↑
Column or Attribute

Schema Example



<https://database.guide/what-is-a-database-schema/>

Vocabulary

- ▶ Data Definition Language (DDL):
 - ▶ CREATE, DROP, ALTER, TRUNCATE
- ▶ Data Manipulation Language (DML):
 - ▶ INSERT, UPDATE, DELETE
- ▶ Data Query Language (DQL):
 - ▶ SELECT, JOIN
- ▶ Data Control Language (DCL):
 - ▶ GRANT, REVOKE

Basic SQL Statements

A screenshot of the DBeaver interface. At the top, there is a script editor window titled "Covid_Data_csv" containing the following SQL code:

```
select * from Covid_Data_csv
select country, population, total_cases from Covid_Data_csv
```

The second query is circled in red. Below it, a results grid window titled "Covid_Data_csv 1" displays data for 181 countries. The columns are "country", "population", and "total_cases". The data starts with Afghanistan, Albania, Algeria, Andorra, Angola, Antigua and Barbuda, Argentina, Armenia, and so on. A large red circle encloses the entire results grid.

JS Writable Smart Insert 3 : 60 : 89



VIEW: virtual table

A screenshot of the DBeaver interface showing a more complex query and its execution plan. The query is:

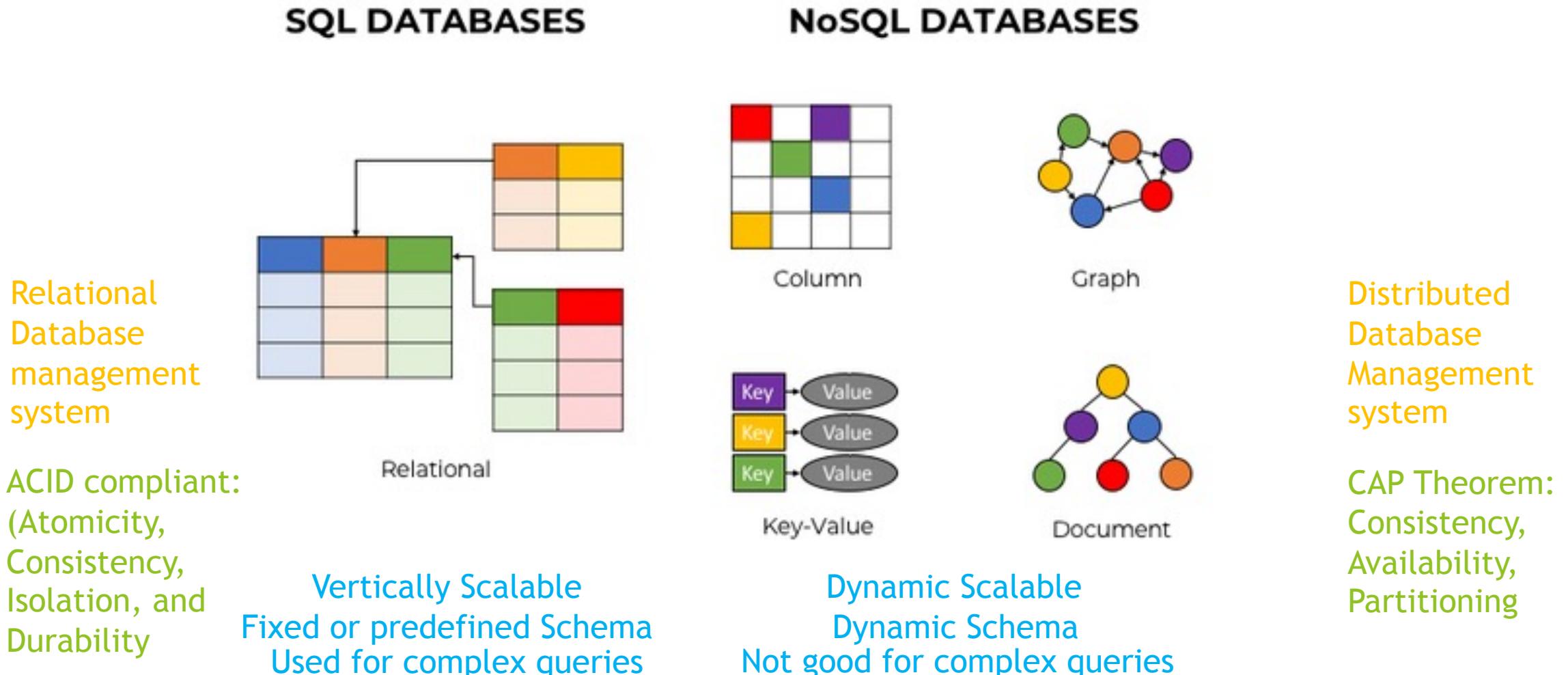
```
select covid_data.country, covid_data.population, covid_data.total_cases, stability_data.stable_inc
from covid_data join stability_data on covid_data.iso_code = stability_data.iso_code
```

An arrow points from the "Results" section of the interface down to the "Execution plan" window, which shows the following details:

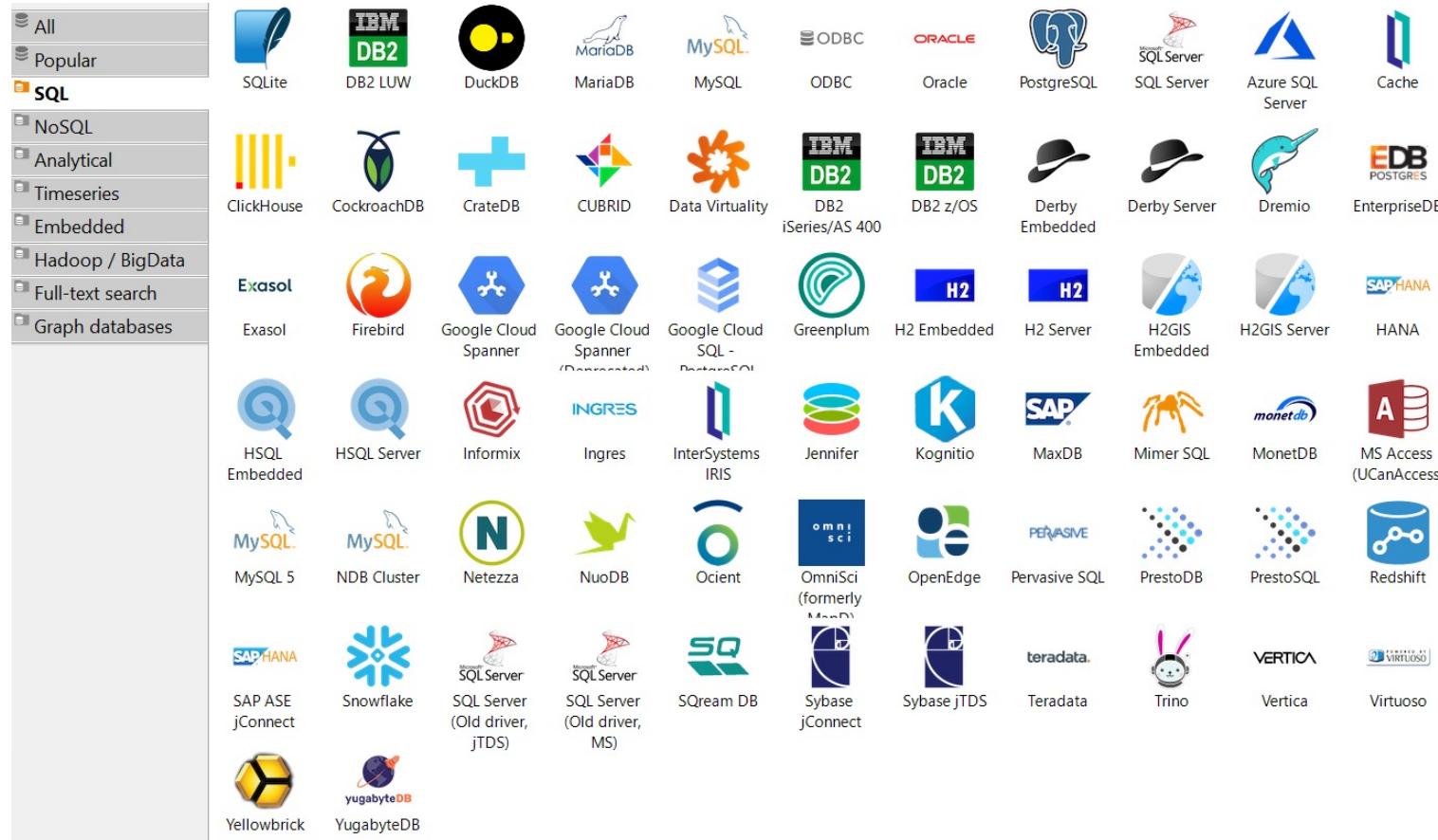
Type	Name	Cost	Rows
select		3,240.12	359
nested_loop#1	stability_data (ALL)	0.25	178
nested_loop#2	covid_data (ALL)	0.27	181



INDEX: users cannot see the index; it is used to retrieve data from the database more quickly



Relational SQL Databases



Interface

Start a new script

Connect Database

Databases and Tables

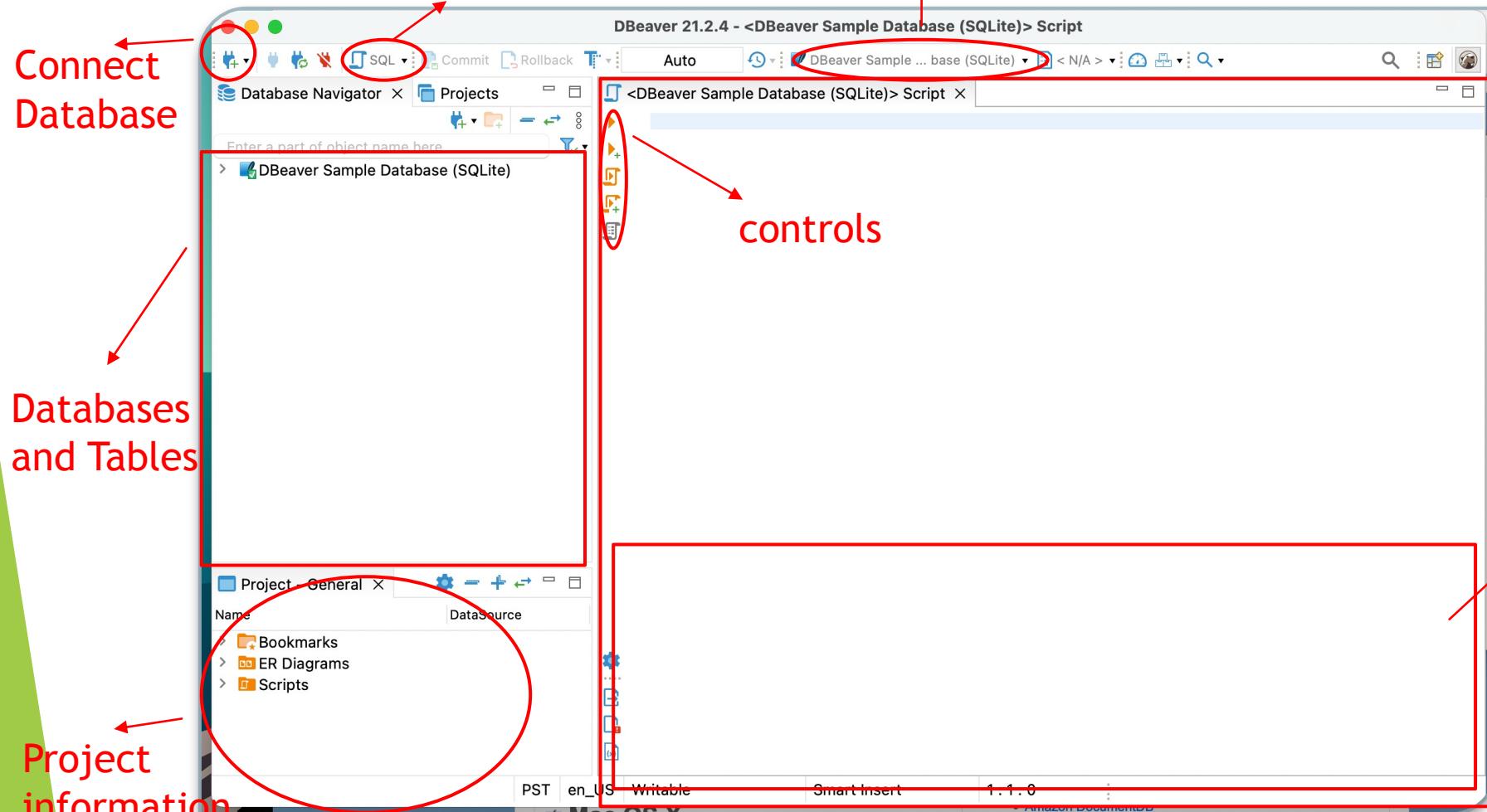
Project information

Choose your data base

controls

Text Editor
Where you will write Queries

View,
Results,
Execution



Today's Goals



Whole - Big Picture

- ▶ Be able to take any data set and import into your database
- ▶ Be able to sort and look through your data set
- ▶ Be able to create your own data set and input data
- ▶ Be able to join data sets together



Parts - Small angles

- ▶ Use basic SQL commands
- ▶ Understand the process of writing a query
- ▶ Understand how data can be joined
- ▶ Understand the behind the scenes of how the query is operating

Today's Data



- ▶ Modified datasets for workshop (4 files total)
- ▶ **State Crime CSV File**
 - ▶ `state_crime.csv`
 - ▶ information on the crime rates and totals for states across the United States for a wide range of years
 - ▶ reports go from 1960 to 2019 (only used 2010, 2014 and 2019)
 - ▶ https://corgis-edu.github.io/corgis/csv/state_crime/
- ▶ **State Demographics CSV and SQL Files**
 - ▶ `state_computer_data.sql`, `state_workforce.csv`, `state_people.sql`
 - ▶ summarized information obtained about states in the United States from 2015 through 2019 through the United States Census Bureau
 - ▶ just the summarized data as of 2019
 - ▶ https://corgis-edu.github.io/corgis/csv/state_demographics/

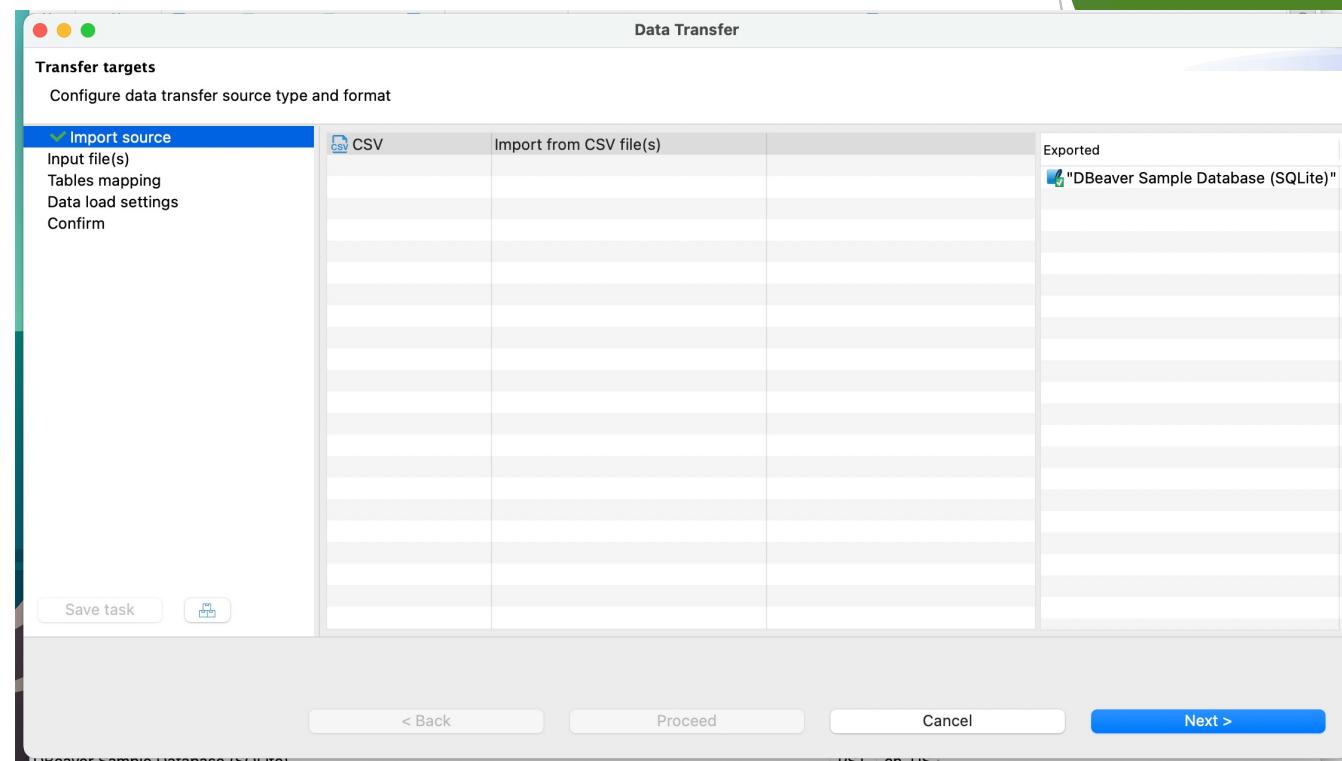
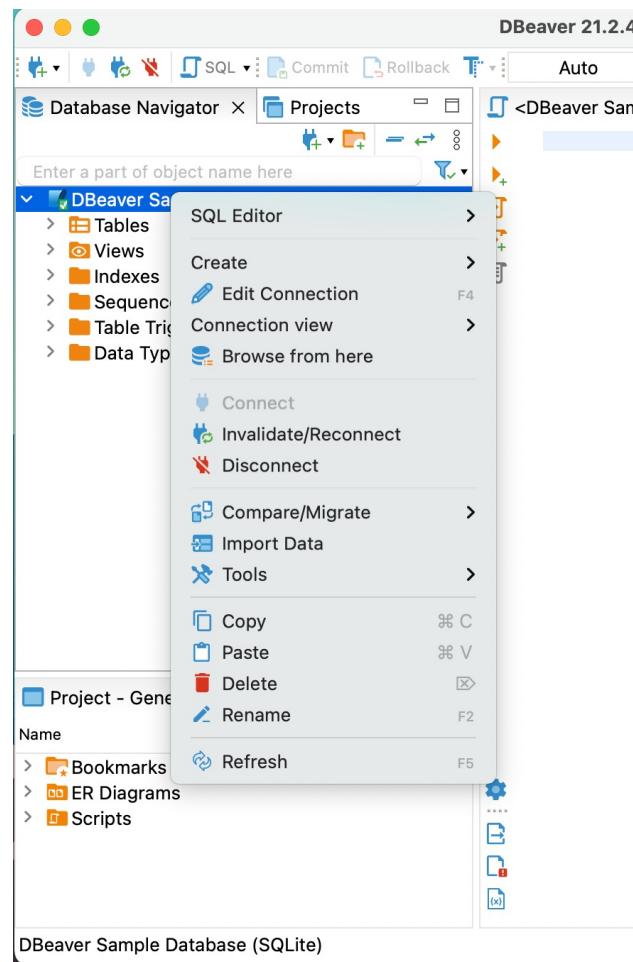
<https://corgis-edu.github.io/corgis/>

Basic SQL Commands

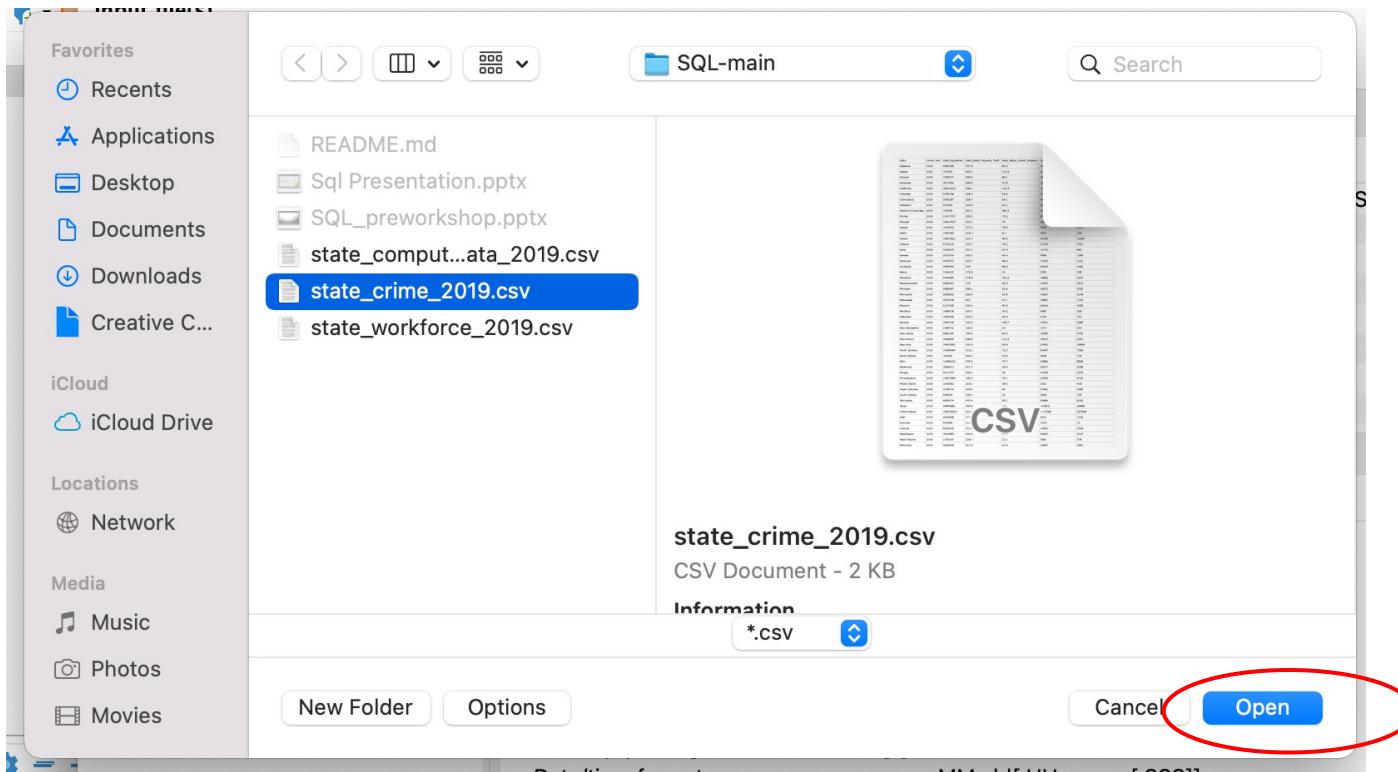
Hands-on

Hands-on Agenda

- ▶ Import a data table called **state_crime**
- ▶ Query using the **SELECT** statement with *
- ▶ Filter a query using the **WHERE** clause and **ORDER BY**
- ▶ Create data table called **state_computer_data**
- ▶ Insert **state_computer_data** from **sql file**
- ▶ Relate the two tables using the **JOIN** connector make **Alias**
- ▶ Use **Group by** to look sort the data set
- ▶ Throughout the workshop: Activities involving files named **state_workforce** and **state_people**

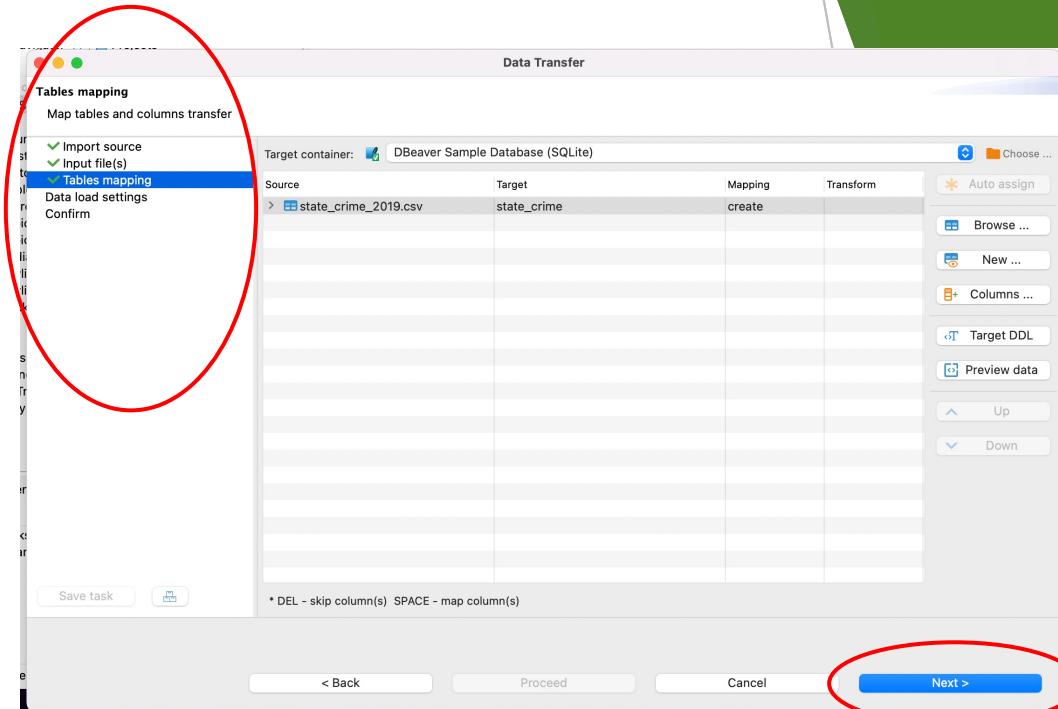
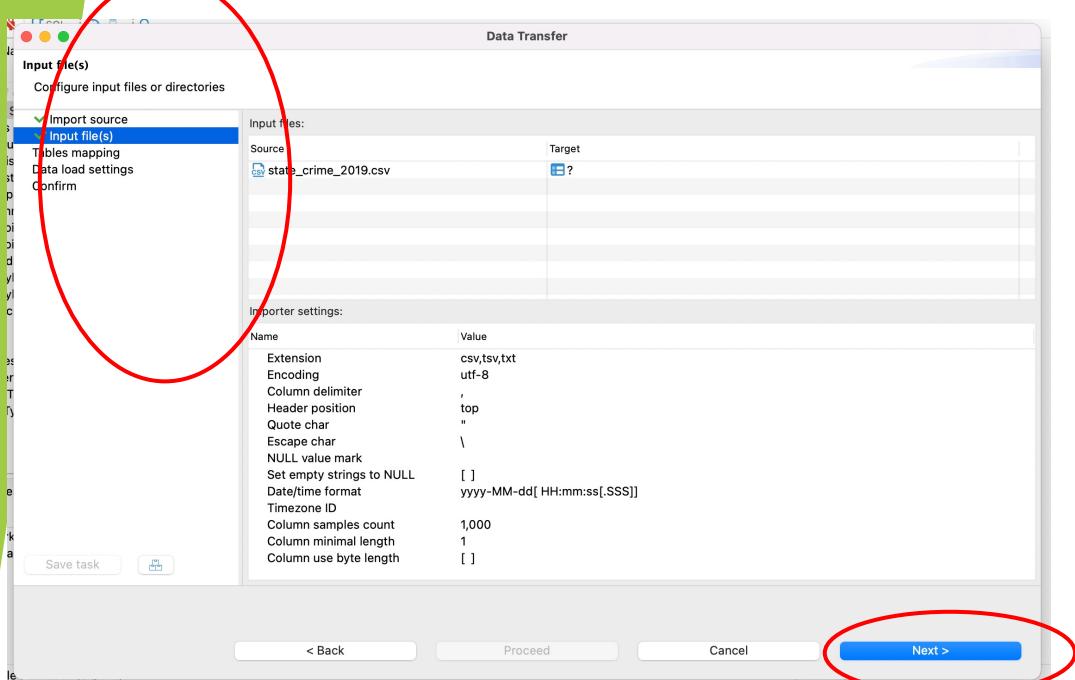


Import CSV

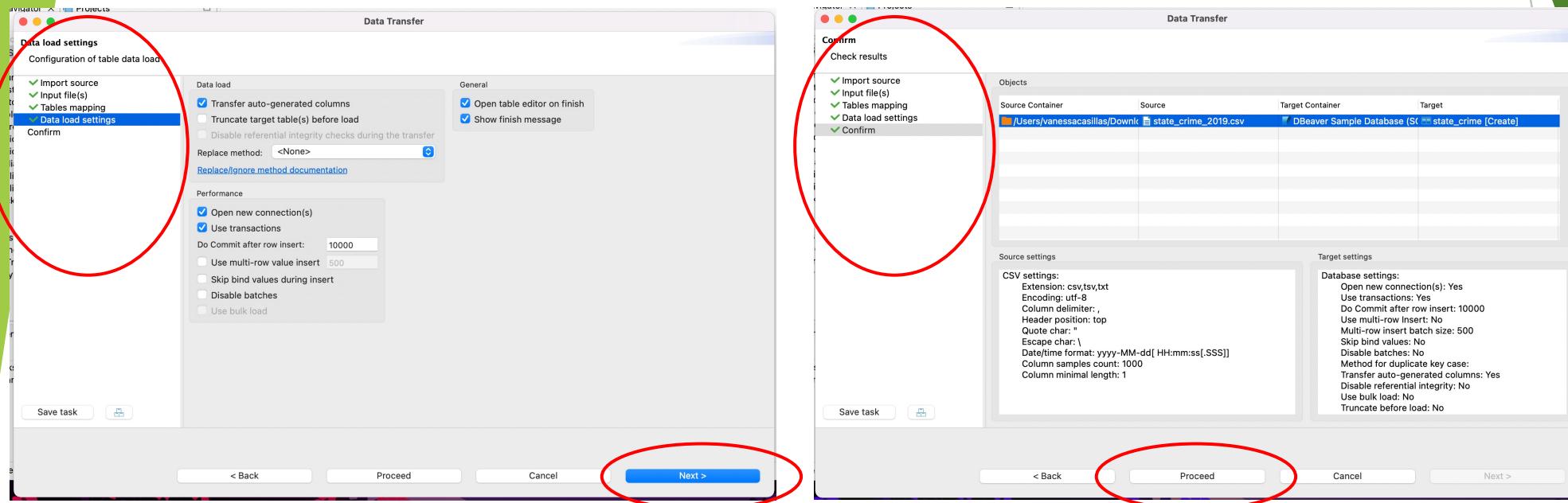


Import CSV (cont.)

Old versions of DBeaver will not see this

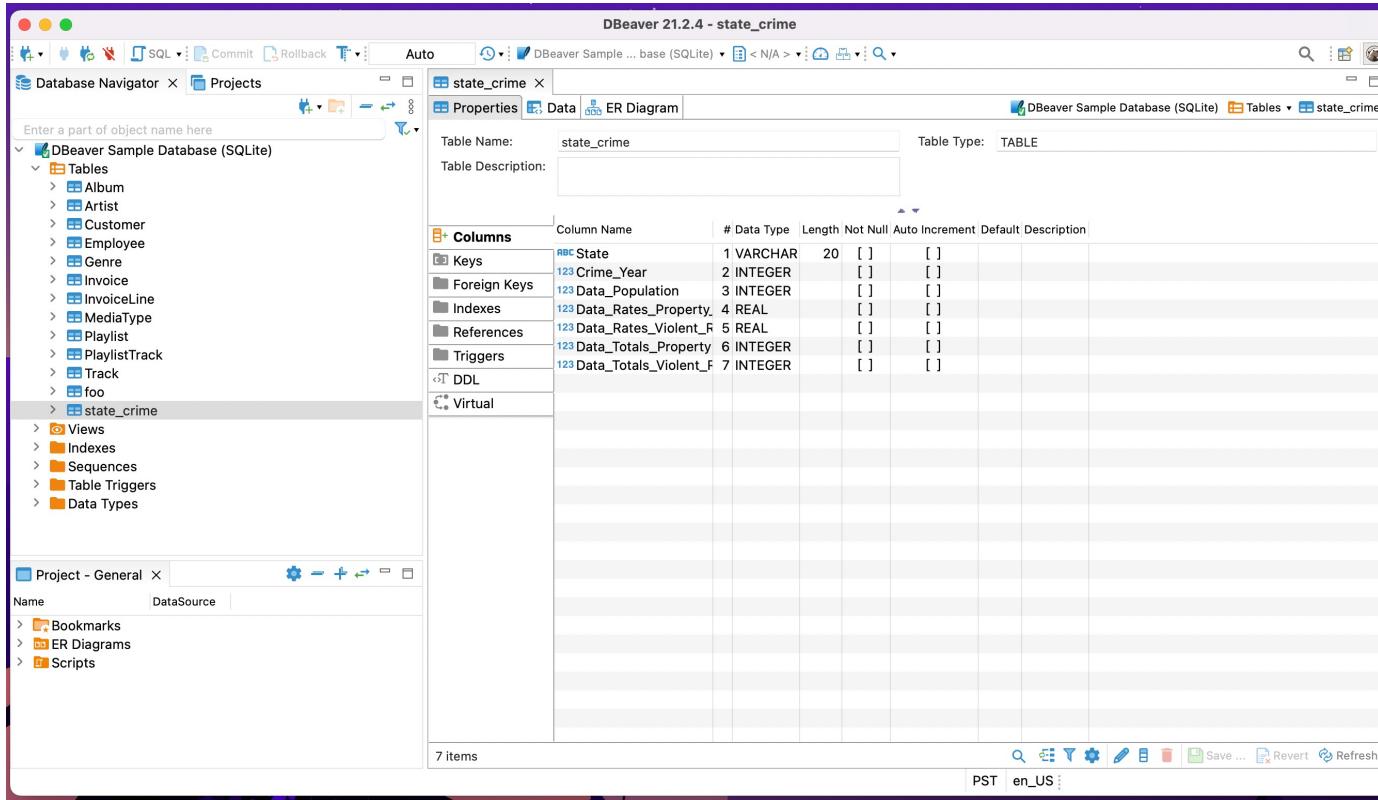


Import CSV (cont.)



Old versions of DBeaver will see start instead

Import CSV (cont.)



Successful Import of CSV

Writing Query

Select - Returns the data that was requested

From - choose a table to draw information from

Join - matches records from different tables

Where - filters data bases on request

Group By - aggregates the data

Having - filers aggregated data

Order by - sorts the data

Limit - limit the number of rows returned

Select

```
select * from state_crime;
```

- ▶ Will show all the data within the table due to the wildcard

Questions

- ▶ How many rows?
- ▶ How many attributes?

Detailed view

```
SELECT column1, column2, ...  
FROM table_name;
```

`*`: wildcard (select all)

Table name

`select * from state_crime;` → syntax

What do you want to see?

what table?

The diagram illustrates the components of the SQL query. A green arrow points from the text 'wildcard (select all)' to the asterisk (*) in the query. A purple arrow points from 'Table name' to the table name 'state_crime'. A blue arrow points from 'syntax' to the entire query. A red arrow points from 'what table?' to the table name 'state_crime'. A cyan arrow points from 'What do you want to see?' to the asterisk (*) in the query.

Select

```
select Data_Population,  
Data_Totals_Property_Theft,  
Data_Totals_Violent_Robbery  
from state_crime;
```

- ▶ Will show the columns that we like to see only

Detailed view

```
SELECT column1, column2, ...  
FROM table_name;
```

What do you want to see?

```
select Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery  
from state_crime; —→ syntax
```

what table?

Table name

Column names



Activity #1 - Select

- ▶ Import the file named state_workforce
 1. How many rows and attributes does this table have?

Where Clause

```
select Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery  
from state_crime  
where Data_Totals_Violent_Robbery >=3000;
```

Detailed view

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

What do you want to see?

Column names

what table?

```
select Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery
from state_crime
where Data_Totals_Violent_Robbery >=3000;
```

Clause: to filter

Condition

The screenshot shows the DBeaver interface with the following details:

- SQL Editor:** The title bar says "<DBeaver Sample Database (SQLite)> QCLworkshop.sql x". The code pane contains the following SQL query:

```
select * from state_crime
select Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery from state_crime
=select Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery
from state_crime where Data_Totals_Violent_Robbery >=3000
```
- Grid Viewer:** The title bar says "state_crime 1 x". It displays the results of the last query in a tabular format. The columns are labeled "Data_Population", "Data_Totals_Property_Theft", and "Data_Totals_Violent_Robbery". The data consists of 14 rows of numerical values.
- Bottom Status Bar:** Shows "26 row(s) fetched - 5ms (+1ms)".
- Bottom Navigation:** Includes buttons for Save, Cancel, Script, and various navigation icons.
- Toolbar:** On the left side, there is a vertical toolbar with icons for settings, database, table, and record selection.

Order by

```
select State, Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery, Crime_Year  
from state_crime  
where Data_Totals_Violent_Robbery >= 3000  
order by Data_Population DESC;
```

- ▶ Default is asc

Detailed view

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
ORDER BY column1, column2, ... ASC|DESC;
```

What do you want to see?

Column names

what table?

```
select State, Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery, Crime_Year
```

← from state_crime → Table name

where Data_Totals_Violent_Robbery >=3000 → Column and Condition

order by Data_Population DESC ; → syntax

Sort results in ascending or descending order

Clause: to filter

state_crime * <DBeaver Sample Database (SQLite)> QCLworkshop.sql

```
select * from state_crime
select Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery from state_crime
select Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery
from state_crime where Data_Totals_Violent_Robbery >=3000
select State, Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery
from state_crime
where Data_Totals_Violent_Robbery >=3000
order by Data_Population DESC
```

state_crime 1

Enter a SQL expression to filter results (use Ctrl+Space)

	State	Data_Population	Data_Totals_Property_Theft	Data_Totals_Violent_Robbery
1	United States	328,239,523	1,117,696	267,988
2	California	39,512,223	152,555	52,301
3	Texas	28,995,881	113,902	28,988
4	Florida	21,477,737	63,396	16,217
5	New York	19,453,561	27,600	18,068
6	Pennsylvania	12,801,989	23,354	9,743
7	Illinois	12,671,821	34,433	12,464
8	Ohio	11,689,100	43,894	8,846
9	Georgia	10,617,423	39,506	7,961
10	North Carolina	10,488,084	54,447	7,599
11	Michigan	9,986,857	28,572	5,350
12	New Jersey	8,882,190	16,399	5,730
13	Virginia	8,535,519	13,900	3,524
14	Washington	7,614,893	34,540	5,147

Grid Text Record

Save Cancel Script | 26 row(s) fetched - 2ms (+1ms)

PST en_US Writable Smart Insert 8 : 1 [177] Sel: 177 | 4 ...



Activity #2 - Where Clause and Order by

- ▶ Use where clause to find out how many people on average take longer than 20 mins to get to work?
- ▶ Use Order by to find out what state has the longest on average time it takes to get to work?

Group By

```
select State,  
Data_Rates_Property_Theft,  
Data_Totals_Violent_Robbery,  
Data_Population  
from state_crime_csv  
where Data_Totals_Violent_Robbery > 10000  
group by State  
order by Data_Totals_Violent_Robbery DESC;
```

Detailed view

```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
GROUP BY column_name(s)  
ORDER BY column_name(s);
```

What do you want to see?

```
select State,  
Data_Rates_Property_Theft,  
Data_Totals_Violent_Robbery,  
Data_Population
```

Column names

```
from state_crime_csv
```

```
where Data_Totals_Violent_Robbery > 10000
```

```
group by State
```

```
order by Data_Totals_Violent_Robbery DESC;
```

Clause: to filter what table?

Puts rows that are similar together

→ Table name

→ Column and Condition

→ syntax

→ Column to sort by

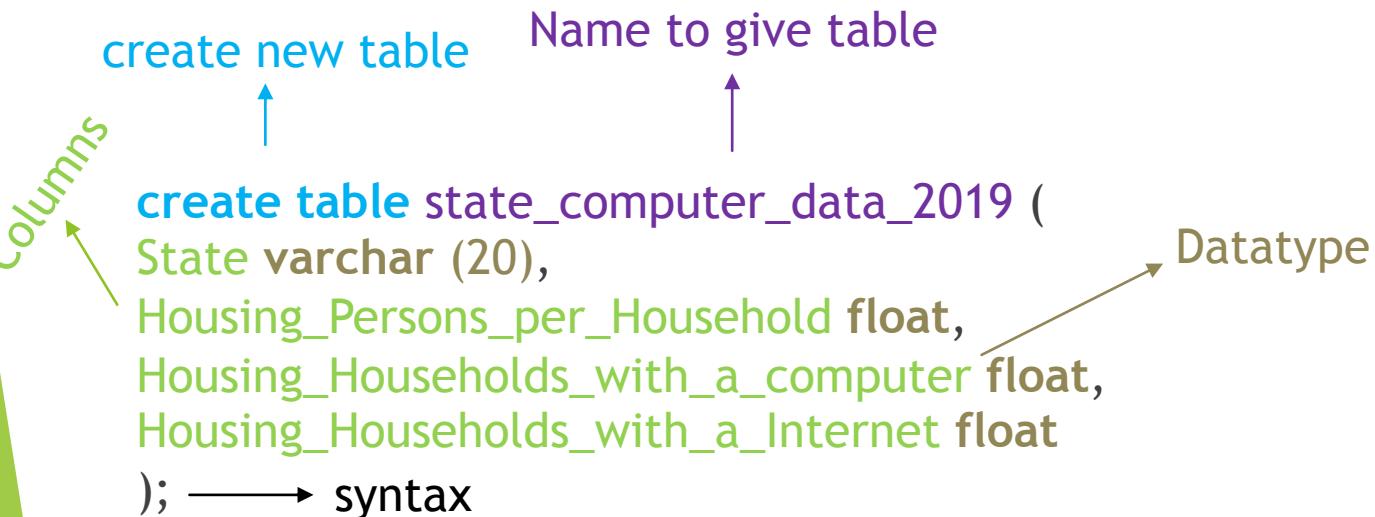
Sort results in ascending or descending order

Create Table

```
create table state_computer_data_2019(  
State Varchar (20),  
Housing_Persons_per_Household Float,  
Housing_Households_with_a_computer Float,  
Housing_Households_with_a_Internet Float  
);
```

Detailed view

```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ...  
)
```

create new table Name to give table

create table state_computer_data_2019 (
State varchar (20),
Housing_Persons_per_Household float,
Housing_Households_with_a_computer float,
Housing_Households_with_a_Internet float
); —→ syntax

state_crime *<DBeaver Sample Database (SQLite)> QCLworkshop.sql

```
select * from state_crime
select Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery from state_crime
select Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery
from state_crime where Data_Totals_Violent_Robbery >=3000
select State, Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery
from state_crime
where Data_Totals_Violent_Robbery >=3000
order by Data_Population DESC
create table state_computer_data_2019
(
State Varchar (20),
Housing_Persons_per_Household Float,
Housing_Households_with_a_computer Float,
Housing_Households_with_a_Internet Float
)
```

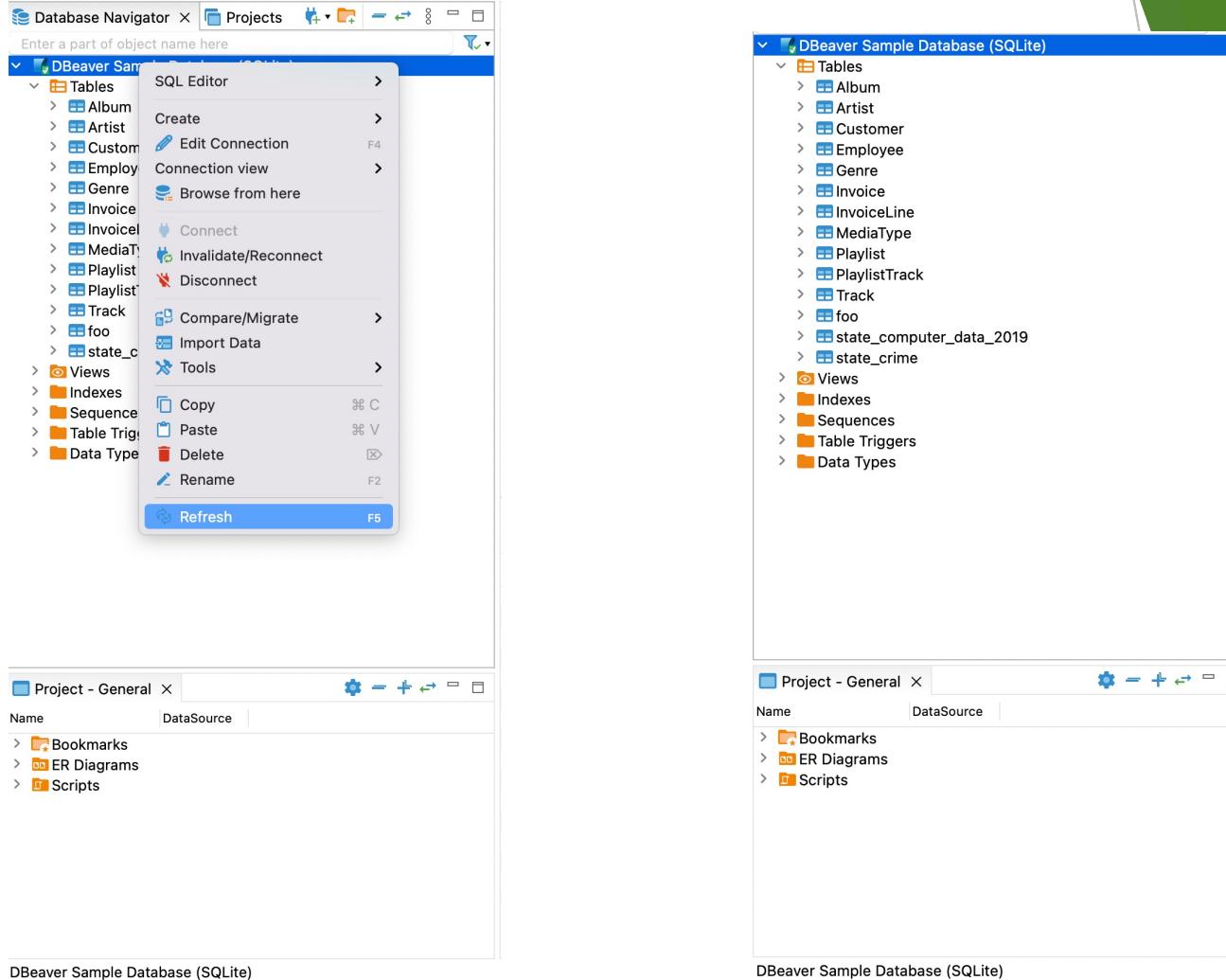
Statistics 1

create table state_computer_data_2019 Enter a SQL expression to filter results (use Ctrl+Space)

Name	Value
Updated Rows	1
Query	create table state_computer_data_2019 (State Varchar (20), Housing_Persons_per_Household Float, Housing_Households_with_a_computer Float, Housing_Households_with_a_Internet Float)
Finish time	Sat Nov 06 18:55:02 PDT 2021

Save Cancel Script | 1 row(s) updated - 4ms | Rows: 1 | 200 | 13 : 1 [184] | Sel: 184 | 7 |

Refresh



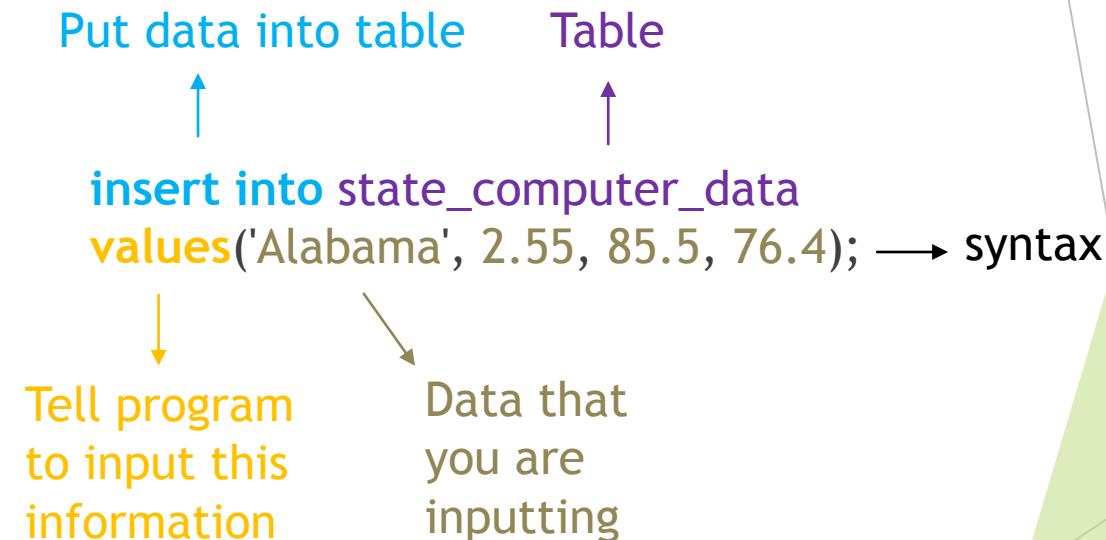
Insert and Detail View

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

```
insert into state_computer_data  
Values('Alabama', 2.55, 85.5, 76.4);
```

```
select *  
from state_computer_data;
```

(breakdown in section of select statement)



state_crime * <DBeaver Sample Database (SQLite)> QCLworkshop.sql X

```

select * from state_crime
select Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery from state_crime
select Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery
from state_crime where Data_Totals_Violent_Robbery >=3000
select State, Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery
from state_crime
where Data_Totals_Violent_Robbery >=3000
order by Data_Population DESC

create table state_computer_data_2019
(
State Varchar (20),
Housing_Persons_per_Household Float,
Housing_Households_with_a_computer Float,
Housing_Households_with_a_Internet Float
)

insert into state_computer_data_2019
Values('Alabama', 2.55, 85.5, 76.4)

```

Statistics 1 X

Insert into state_computer_data_2019 Enter a SQL expression to filter results (use Ctrl+Space)

Name	Value
Updated Rows 1	
Query	insert into state_computer_data_2019 Values('Alabama', 2.55, 85.5, 76.4)
Finish time	Sat Nov 06 18:56:30 PDT 2021

1 row(s) updated - 5ms

PST en_US Writable Smart Insert 21:1 [73] Sel: 73 | 2 ...

state_crime * <DBeaver Sample Database (SQLite)> QCLworkshop.sql X

```

select * from state_crime
select Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery from state_crime
select Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery
from state_crime where Data_Totals_Violent_Robbery >=3000
select State, Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery
from state_crime
where Data_Totals_Violent_Robbery >=3000
order by Data_Population DESC

create table state_computer_data_2019
(
State Varchar (20),
Housing_Persons_per_Household Float,
Housing_Households_with_a_computer Float,
Housing_Households_with_a_Internet Float
)

insert into state_computer_data_2019
Values('Alabama', 2.55, 85.5, 76.4)
select * from state_computer_data_2019
```

state_computer_data_2019 1 X

Select * from state_computer_data_2019 Enter a SQL expression to filter results (use Ctrl+Space)

State	Housing_Persons_per_Household	Housing_Households_with_a_computer	Housing_Households_with_a_Internet
Alabama	2.55	85.5	76.4

1 row(s) fetched - 3ms (+3ms)

PST en_US Writable Smart Insert 24:39:743 Sel: 0 | 0 ...

Insert data (alternative)



- ▶ Load state_computer_data.sql
 - ▶ Click on data source
 - ▶ Click select

The screenshot shows the DBeaver interface with multiple tabs open. The main tab displays an SQL script titled 'state_crime' with the following content:

```
INSERT INTO state_computer_data (State,Housing_Persons_per_Household,Housing_Households_with_a_computer,Housing_Households_with_a_Internet)
VALUES('Alaska',2.8,94.1,85.5),
      ('Arizona',2.68,91.7,84.1),
      ('Arkansas',2.52,86.2,73.0),
      ('California',2.95,93.0,86.7),
      ('Colorado',2.56,93.9,87.6),
      ('Connecticut',2.53,90.8,85.5),
      ('Delaware',2.57,91.6,85.0),
      ('District of Columbia',2.3,91.8,82.6),
      ('Florida',2.65,91.5,83.0),
      ('Georgia',2.7,90.2,81.3),
      ('Hawaii',3.0,91.2,84.8),
      ('Idaho',2.68,91.8,82.7),
      ('Illinois',2.57,89.9,82.7),
      ('Indiana',2.52,86.7,80.1),
      ('Iowa',2.6,89.4,81.0),
      ('Kansas',2.51,90.0,81.8),
      ('Kentucky',2.49,86.4,78.4),
      ('Louisiana',2.61,85.6,75.5),
      ('Maine',2.46,93.0,82.5),
      ('Maryland',2.67,92.5,86.4),
      ('Massachusetts',2.52,91.4,86.4),
      ('Michigan',2.47,89.6,81.5),
      ('Minnesota',2.49,91.6,84.8),
      ('Mississippi',2.42,83.8,74.15),
      ('Missouri',2.46,90.4,81.2),
      ('Montana',2.39,88.9,80.7),
      ('Nebraska',2.45,90.0,83.4),
      ('Nevada',2.67,92.5,83.2),
      ('New Hampshire',2.46,93.0,87.7),
      ('New Jersey',2.6,89.4,81.8),
      ('New Mexico',2.63,85.9,74.6),
      ('New York',2.59,89.6,82.8),
      ('North Carolina',2.52,89.1,80.7)
```

Below the script, a 'Statistics 1' panel shows the query and its results:

Name	Value
Updated Rows	50
Query	INSERT INTO state_computer_data (State,Housing_Persons_per_Household,Housing_Households_with_a_computer,Housing_Households_with_a_Internet) VALUES('Alaska',2.8,94.1,85.5), ('Arizona',2.68,91.7,84.1), ('Arkansas',2.52,86.2,73.0), ('California',2.95,93.0,86.7), ('Colorado',2.56,93.9,87.6), ('Connecticut',2.53,90.8,85.5), ('Delaware',2.57,91.6,85.0), ('District of Columbia',2.3,91.8,82.6), ('Florida',2.65,91.5,83.0), ('Georgia',2.7,90.2,81.3), ('Hawaii',3.0,91.2,84.8), ('Idaho',2.68,91.8,82.7), ('Illinois',2.57,89.9,82.7), ('Indiana',2.52,86.7,80.1), ('Iowa',2.6,89.4,81.0), ('Kansas',2.51,90.0,81.8), ('Kentucky',2.49,86.4,78.4), ('Louisiana',2.61,85.6,75.5), ('Maine',2.46,93.0,82.5), ('Maryland',2.67,92.5,86.4), ('Massachusetts',2.52,91.4,86.4), ('Michigan',2.47,89.6,81.5), ('Minnesota',2.49,91.6,84.8), ('Mississippi',2.42,83.8,74.15), ('Missouri',2.46,90.4,81.2), ('Montana',2.39,88.9,80.7), ('Nebraska',2.45,90.0,83.4), ('Nevada',2.67,92.5,83.2), ('New Hampshire',2.46,93.0,87.7), ('New Jersey',2.6,89.4,81.8), ('New Mexico',2.63,85.9,74.6), ('New York',2.59,89.6,82.8), ('North Carolina',2.52,89.1,80.7)

At the bottom, it says '50 row(s) updated - 18ms'.

Insert data (cont.)

select * from state_computer_data;

(breakdown in section of select statement)

Look at table data

Which column looks like a column that has the same information for both tables?

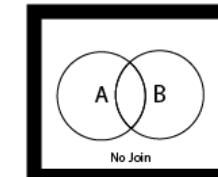
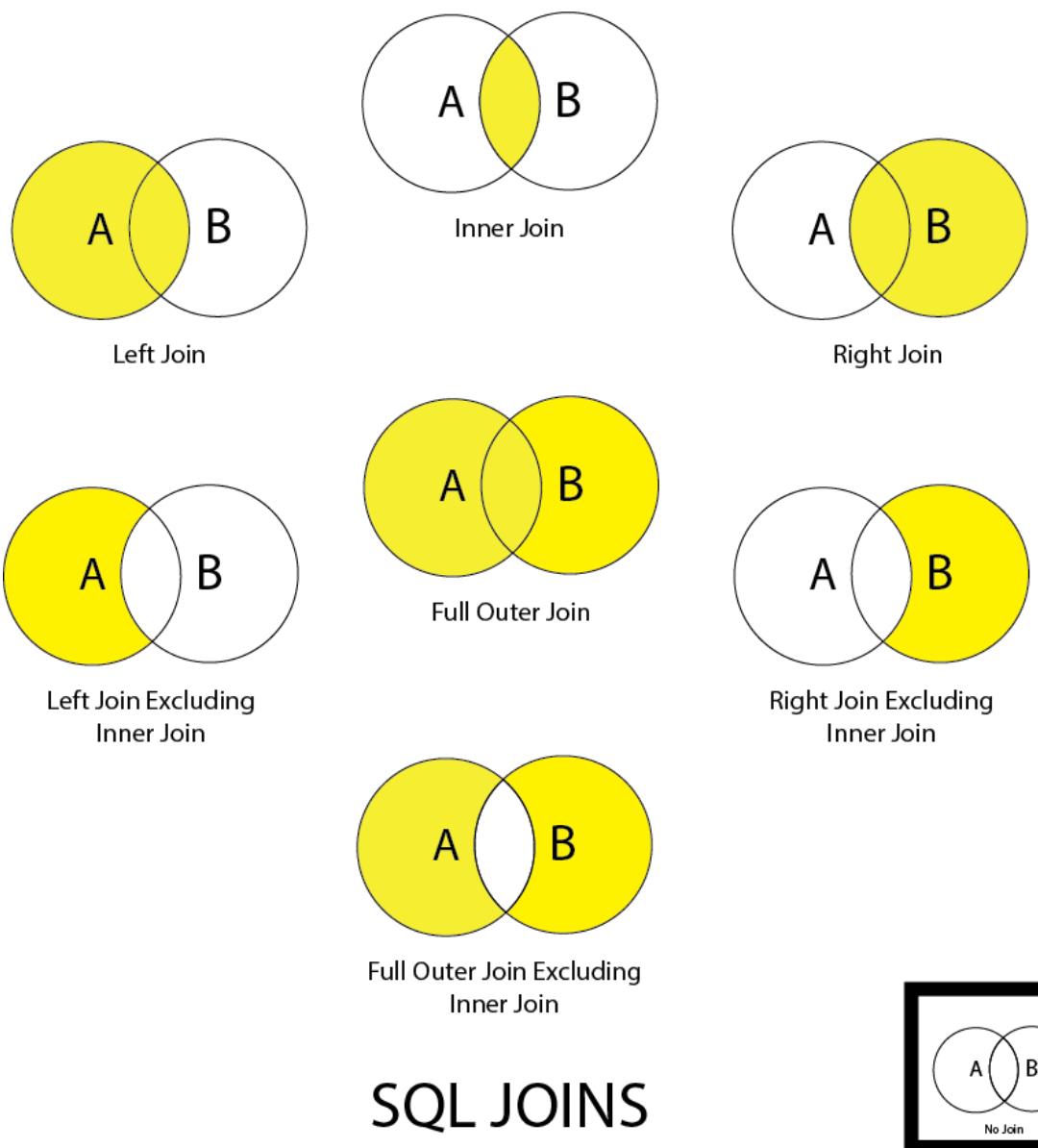
The screenshot shows the DBeaver interface with two panes. The top pane contains an SQL script with several queries and a table creation statement for 'state_computer_data_2019'. The bottom pane displays a grid view of the 'state_computer_data_2019' table, which has 13 rows and 5 columns: State, Housing_Persons_per_Household, Housing_Households_with_a_computer, Housing_Households_with_a_Internet, and a primary key column (labeled '1'). The data includes states like Alabama, Alaska, Arizona, etc., with their respective household statistics.

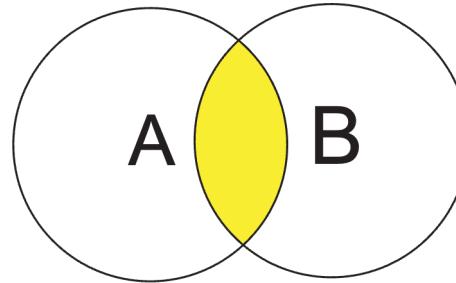
	State	Housing_Persons_per_Household	Housing_Households_with_a_computer	Housing_Households_with_a_Internet
1	Alabama	2.55	85.5	76.4
2	Alaska	2.8	94.1	85.5
3	Arizona	2.68	91.7	84.1
4	Arkansas	2.52	86.2	73
5	California	2.95	93	86.7
6	Colorado	2.56	93.9	87.6
7	Connecticut	2.53	90.8	85.5
8	Delaware	2.57	91.6	85
9	District of Col	2.3	91.8	82.6
10	Florida	2.65	91.5	83
11	Georgia	2.7	90.2	81.3
12	Hawaii	3	91.2	84.8
13	Idaho	2.68	91.8	82.7



Activity #3 - Create Table and Insert Data

- ▶ Create a table named state_people and add the attributes:
State as a varchar with 20 characters,
Employment_Firms_Total as a integer,
Employment_Firms_Men_Owned as a integer,
Employment_Firms_Women_Owned as a integer,
Age_Percent_Under_18_Years as a float,
and Age_Percent_65_and_Older as a float.
- ▶ Insert data into your new table called state_people using a sql file in the downloaded github folder





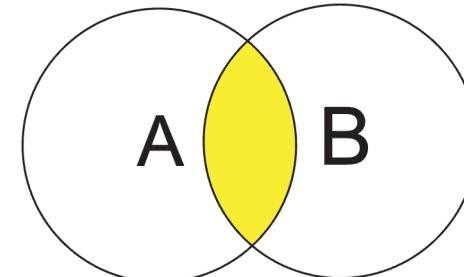
Inner Join

Inner Join

```
select state_computer_data.State,  
state_computer_data_2019.Housing_Households_with_a_computer,  
state_crime.Data_Totals_Property_Theft  
from state_computer_data  
join state_crime  
on state_computer_data.state = state_crime.State;
```

Detail View

```
SELECT column_name(s)  
FROM table1  
INNER JOIN table2  
ON table1.column_name = table2.column_name;
```



Inner Join

What do you
want to see?

Table name

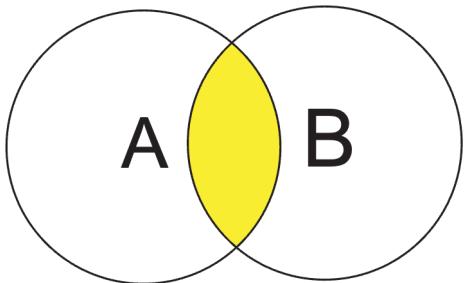
Column names

```
select state_computer_data.State,  
state_computer_data.Housing_Households_with_a_computer,  
state_crime.Data_Totals_Property_Theft  
← from state_computer_data  
join state_crime  
on state_computer_data.state = state_crime.State; → syntax
```

what table?

What table are you joining and on what columns?

You can just join if you are making an inner join, yet if you are
making any other join, you do have to specify



Inner Join

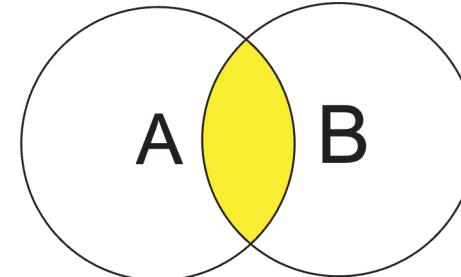
```

state_crime  *<DBeaver Sample Database (SQLite)> QCLworkshop.sql ×  *<DBeaver Sample Database (SQLite)> state_computer_data.s
▶   select * from state_crime
▶+
▶   select Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery from state_crime
▶
▶   select Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery
▶   from state_crime where Data_Totals_Violent_Robbery >=3000
▶
▶   select State, Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery
▶   where Data_Totals_Violent_Robbery >=3000
▶   order by Data_Population DESC
▶
▶   create table state_computer_data_2019
▶   (
▶       State Varchar (20),
▶       Housing_Persons_per_Household Float,
▶       Housing_Households_with_a_computer Float,
▶       Housing_Households_with_a_Internet Float
▶   )
▶
▶   insert into state_computer_data_2019
▶   Values('Alabama', 2.55, 85.5, 76.4)
▶
▶   select * from state_computer_data_2019
▶
▶   select state_computer_data_2019.State, state_computer_data_2019.Housing_Households_with_a_computer, state_crime.Data_Totals_Property_Theft
▶   from state_computer_data_2019
▶   join state_crime on state_computer_data_2019.state = state_crime.State;
▶
state_computer_data_2019(+) 1 ×
select state_computer_data_2019.St Enter a SQL expression to filter results (use Ctrl+Space)
Grid State Housing_Households_with_a_computer Data_Totals_Property_Theft
Text
Record
1 Alabama 85.5 26,079
2 Alaska 94.1 3,563
3 Arizona 91.7 28,699
4 Arkansas 86.2 18,095
5 California 93 152,555
6 Colorado 93.9 20,064
7 Connecticut 90.8 6,441
8 Delaware 91.6 2,968
9 District of Columbia 91.8 1,843
10 Florida 91.5 63,396
11 Georgia 90.2 39,506
12 Hawaii 91.2 5,340
13 Idaho 91.8 3,927
Panels
Save Cancel Script 51 row(s) fetched - 2ms
PST en_US Writable Smart Insert 26 : 1 [243] Sel: 243 | 3 ...

```

Alias

```
select a.State,  
a.Housing_Households_with_a_computer,  
b.Data_Totals_Property_Theft,  
b.Data_Totals_Violent_Robbery  
from state_computer_data_2019 as a  
join state_crime as b  
on a.state = b.State;
```



Inner Join

Detail View

```
SELECT column_name(s)  
FROM table_name AS alias_name;
```

what table?

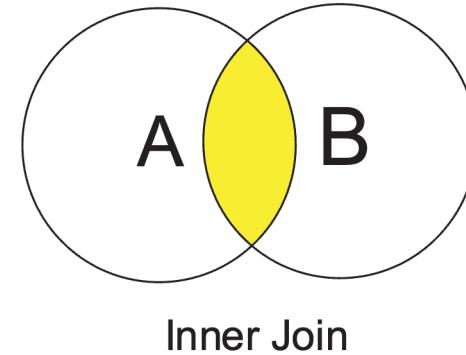
What do you
want to see?
Table name

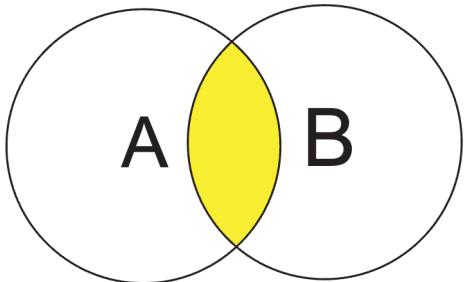
```
select a.State,  
a.Housing_Households_with_a_computer,  
b.Data_Totals_Property_Theft,  
b.Data_Totals_Violent_Robbery  
from state_computer_data as a  
join state_crime as b  
on a.state = b.State; —→ syntax
```

Column names

What table are you joining and on what columns?

You can just join if you are making an inner join, yet if you are
making any other join, you do have to specify





Inner Join

state_crime * <DBeaver Sample Database (SQLite)> QCLworkshop.sql X * <DBeaver Sample Database (SQLite)> state_computer_data.s

```

    select Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery
    from state_crime where Data_Totals_Violent_Robbery >=3000

    select State, Data_Population, Data_Totals_Property_Theft, Data_Totals_Violent_Robbery
    from state_crime
    where Data_Totals_Violent_Robbery >=3000
    order by Data_Population DESC

    create table state_computer_data_2019
    (
    State Varchar (20),
    Housing_Persons_per_Household Float,
    Housing_Households_with_a_computer Float,
    Housing_Households_with_a_Internet Float
    )

    insert into state_computer_data_2019
    Values('Alabama', 2.55, 85.5, 76.4)

    select * from state_computer_data_2019

    select state_computer_data_2019.State, state_computer_data_2019.Housing_Households_with_a_computer, state_crime.Data_Totals_Property_Theft
    from state_computer_data_2019
    join state_crime on state_computer_data_2019.state = state_crime.State;

    select a.State, a.Housing_Households_with_a_computer, b.Data_Totals_Property_Theft, b.Data_Totals_Violent_Robbery
    from state_computer_data_2019 as a
    join state_crime as b on a.state = b.State;

```

state_computer_data_2019(+) 1 ×

select a.State, a.Housing_Household Enter a SQL expression to filter results (use Ctrl+Space)

	State	Housing_Households_with_a_computer	Data_Totals_Property_Theft	Data_Totals_Violent_Robbery
1	Alabama	85.5	26,079	3,941
2	Alaska	94.1	3,563	826
3	Arizona	91.7	28,699	6,410
4	Arkansas	86.2	18,095	1,557
5	California	93	152,555	52,301
6	Colorado	93.9	20,064	3,663
7	Connecticut	90.8	6,441	1,929
8	Delaware	91.6	2,968	790
9	District of Columbia	91.8	1,843	2,713
10	Florida	91.5	63,396	16,217
11	Georgia	90.2	39,506	7,961
12	Hawaii	91.2	5,340	1,131
13	Idaho	91.8	3,927	155

51 row(s) fetched - 1ms

PST | en_US | Writable | Smart Insert | 30 : 1 [194] | Sel: 194 | 3 |



Activity #4 - Inner Join with Alias

- ▶ Create an inner join using aliases with tables state_workforce and state_people, make sure to view attributes state, Mean_Travel_Time_to_Work, and Employment_Firms_Total

Writing Query

Select - Returns the data that was requested

From - choose a table to draw information from

Join - matches records from different tables

Where - filters data bases on request

Group By - aggregates the data

Having - filers aggregated data

Order by - sorts the data

Limit - limit the number of rows returned

Operation Order

From - choose a table to draw information from

Join - matches records from different tables

Where - filters data bases on request

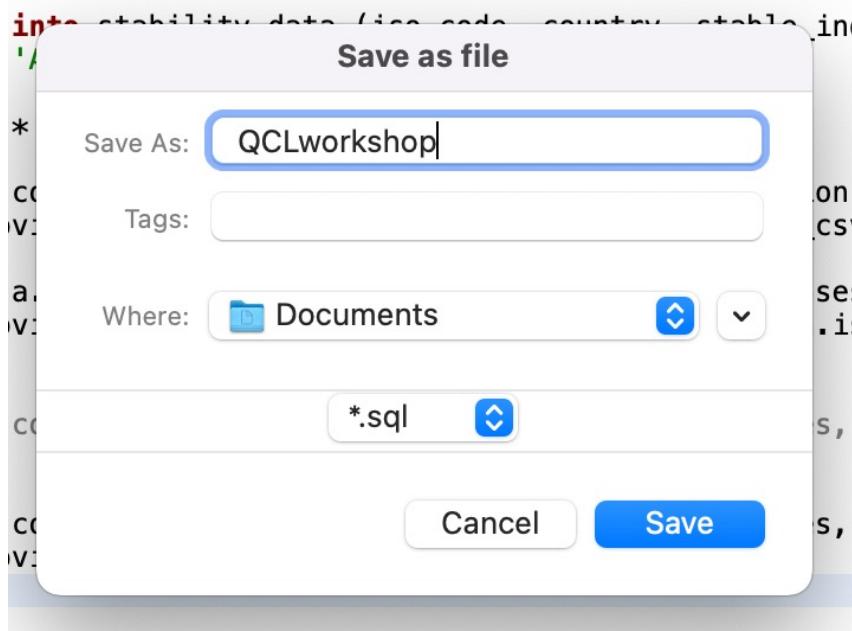
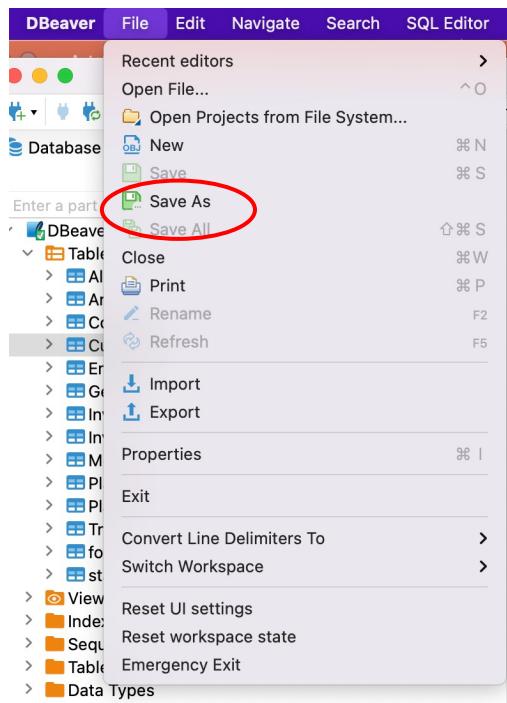
Group By - aggregates the data

Having - filers aggregated data

Select - Returns the data that was requested

Order by - sorts the data

Limit - limit the number of rows returned



Save

Resources

- ▶ Dbeaver Wiki - <https://github.com/dbeaver/dbeaver/wiki>
- ▶ W3schools - <https://www.w3schools.com/sql/default.asp>
- ▶ Code Academy- <https://www.w3schools.com/sql/default.asp>
- ▶ Quizlet - <https://quizlet.com/> (for vocab testing)

Best way to learn

- ▶ SQL Murder Mystery - <https://mystery.knightlab.com/>

Contact info

- ▶ QCL: QCL@cmc.edu
- ▶ Vanessa Casillas: vanessa.casillas@claremontmckenna.edu